Putting Randomized Compiler Testing into Production (Artifact)

Alastair F. Donaldson
Google, London, United Kingdom
Imperial College London, United Kingdom
afdx@google.com

Hugues Evrard
Google, London, United Kingdom
hevrard@google.com

Paul Thomson
Google, London, United Kingdom
paulthomson@google.com

Abstract

This artifact accompanies our experience report for our compiler testing technology transfer project: taking the GraphicsFuzz research project on randomized metamorphic testing of graphics shader compilers, and building the necessary tooling around it to provide a highly automated process for improving the Khronos Vulkan Conformance Test Suite (CTS) with test cases that expose fuzzer-found compiler bugs, or that plug gaps in test coverage. The artifact consists of two Dockerfiles and associated files that can be used to build two Docker containers. The containers include our main tool for performing fuzzing: gfauto. The containers allow the user to fuzz SwiftShader, a software Vulkan implementation, finding 4 bugs. The user will also perform some line coverage analysis of SwiftShader using our tools to synthesize a small test that increases line coverage. Ubuntu, gfauto, SwiftShader, and other dependencies inside the Docker containers are fixed at specific versions, and all random seeds are set to specific values. Thus, all examples should reproduce faithfully on any machine.

2012 ACM Subject Classification Software and its engineering → Compilers; Software and its engineering → Software testing and debugging

Keywords and phrases Compilers, metamorphic testing, 3D graphics, experience report

Digital Object Identifier 10.4230/DARTS.6.2.3

Acknowledgements We are grateful to the anonymous ECOOP 2020 artifact reviewers for their feedback.

Related Article Alastair F. Donaldson, Google, London,UK, and Paul Thomson, "Putting Randomized Compiler Testing into Production", in 34th European Conference on Object-Oriented Programming (ECOOP 2020), LIPIcs, Vol. 166, pp. 22:1–22:29, 2020.
`https://doi.org/10.4230/LIPIcs.ECOOP.2020.22`

Related Conference 34th European Conference on Object-Oriented Programming (ECOOP 2020), November 15–17, 2020, Berlin, Germany (Virtual Conference)

1 Scope

The artifact validates the following claims from our experience report, which are also enumerated (with instructions) in the `README.md` file in our artifact. We reference the relevant section from our experience report in each case. Section 5 is addressed last because it uses a separate Docker container.

- We can find bugs through cross-compilation (Section 3.1)
- We support crash and wrong image tests (Sections 3.3 and 3.4)
- We support loop limiters and array bounds clamping (Section 3.5)
- We can replay self-contained tests using gfauto (Section 4.1)

- Bugs are de-duplicated into crash buckets (Section 4.2)
- Bugs found in previous fuzzing runs are ignored (Section 4.2)
- Vulkan CTS test export (Section 4.3)
- We find bugs in the SPIR-V tooling ecosystem (Section 6)
- We can do differential coverage analysis (Section 5.2)
- We can automatically synthesize small tests that fill coverage gaps (Section 5.3)

2 Content

The artifact package includes:

- `README.md`: the instructions in Markdown format.
- `bug_image/Dockerfile`: the first Dockerfile for validating all claims except for coverage analysis claims.
- `bug_image/*/**`: associated files for building the first Docker container, including a corpus of input files for the fuzzer and an example bug report.
- `coverage_image/Dockerfile`: the second Dockerfile for validating all coverage analysis claims.
- `coverage_image/*/**`: associated files for building the second Docker container, including a corpus of input files for the fuzzer and an example patch for SwiftShader.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `http://multicore.doc.ic.ac.uk/tools/GraphicsFuzz/ECOOP2020Artifact/`. All our tools are open source and available at: `https://github.com/google/graphicsfuzz`.

4 Tested platforms

The artifact requires Docker and network access to build and run the Docker containers. It has been tested on Linux, but should work on any machine or cloud service that supports Docker Linux containers. Note that building the Docker containers can take 1-3 hours, depending on your machine, and approximately 30GB of disk space is required.

5 License

The artifact is available under the Apache 2.0 license: `http://www.apache.org/licenses/LICENSE-2.0`

6 MD5 sum of the artifact

bcc7f8a6d48dd8b2b92625831a435f36

7 Size of the artifact

242 KB